

2. (ONCE AMENDED) A method as recited in claim 1, wherein the algorithm comprises a self evaluating data structure.

B2

3. (ONCE AMENDED) A method as recited in claim 2, wherein the algorithm comprises an algorithm having a defined set and type of inputs and outputs.

B3

5. (ONCE AMENDED) A method as recited in claim 2, wherein the structure comprises an algorithm calling method.

7. (ONCE AMENDED) A method as recited in claim 6, wherein the algorithm parameter types are identified dynamically as the dependency graph is executed.

B4

8. (ONCE AMENDED) A method as recited in claim 7, wherein the data structure contains information describing a set of input and output parameters the algorithm accepts.

9. (ONCE AMENDED) A method as recited in claim 8, wherein the information determines if algorithm attribute types within the dependency graph are compatible.

B5

11. (ONCE AMENDED) A method as recited in claim 1, further comprising mapping parameters of first and second algorithms of the first and second nodes.

B6

15. (ONCE AMENDED) A method as recited in claim 11, wherein the algorithm data structure and value index are passed for the mapping.

B7

17. (TWICE AMENDED) A method, comprising:
evaluating a dependency graph of a graphics creation process
using a computer, comprising:
passing a pointer to an algorithm of a first dependency

node to a second dependency node allowing the second dependency node to execute the algorithm of the dependency node, the algorithm comprising a self evaluating data structure comprising an algorithm calling method and containing information describing a set of input and output parameters the algorithm accepts where the information determines if algorithm attribute types within the dependency graph are compatible and comprising default values for all input and output parameters;

b1 mapping parameters of first and second algorithms of the first and second nodes, where the mapping comprises an index, defines a relationship where input parameters are ignored and output parameters are unmapped and take on default values, where parameter value and type are passed for the mapping and the algorithm data structure and value index are passed for the mapping; and

executing the algorithm of the first dependency node as part of an evaluation of the second dependency node using the pointer and comprising determining a type of a passed parameter where parameter types are identified dynamically as the dependency graph is executed.

19. (TWICE AMENDED) A method, comprising:

evaluating a dependency graph of a graphics creation process using a computer, comprising:

passing a pointer to an algorithm from a first node in a node network to a second node in the node network allowing the second node to execute the algorithm; and

b8 executing the algorithm as part of an evaluation of the second node.

20. (TWICE AMENDED) An apparatus comprising a computer including a dependency node evaluation system having pointers to algorithms passed between nodes of a dependency graph of a graphics creating